MetaLogo: a generator and aligner for multiple sequence logos

Yaowen Chen[a], Zhen He[a], Yahui Men[a], Guohua Dong[a], Shuofeng Hu[a], Xiaomin Ying[a]

[a] Center for Computational Biology, Beijing Institute of Basic Medical Sciences, Beijing, China

Corresponding author: Xiaomin Ying, Center for Computational Biology, Beijing Institute of Basic Medical Sciences, Beijing 100850, China. Email: yingxmbio@foxmail.com

Abstract

Sequence logos are used to visually display sequence conservations and variations. They can indicate the fixed patterns or conserved motifs in a batch of DNA or protein sequences. However, most of the popular sequence logo generators can only draw logos for sequences of the same length, let alone for groups of sequences with different characteristics besides lengths. To solve these problems, we developed MetaLogo, which can draw sequence logos for sequences of different lengths or from different groups in one single plot and align multiple logos to highlight the sequence pattern dynamics across groups, thus allowing users to investigate functional motifs in a more delicate and dynamic perspective. We provide users a public MetaLogo web server (http://metalogo.omicsnet.org), a standalone Python package (https://github.com/labomics/MetaLogo), and also a built-in web server available for local deployment. Using MetaLogo, users can draw informative, customized, aesthetic, and publishable sequence logos without any programming experience.

Keywords

Multiple Sequence logo, Logo alignment, Web server

Introduction

Sequence logo was first proposed by Schneider and Stephens in 1990 [1], and has been widely used thousands of times for sequence pattern visualization in the academic field. Each position of a sequence logo is stacked by different amino acids or nucleotides, with the height of each base indicating its degree of conservation at that position. The most commonly used sequence logo generators include Weblogo [2] , Seq2Logo [3], ggseqlogo [4], Logomaker [5] and RaacLogo [6] and others, involving web servers, Python and R packages, etc. These tools greatly accelerate the researchers' exploration of sequence patterns and motifs.

However, a common problem is that most sequence logo tools only support equal length sequences as input, users need to select the most representative length of sequences to filter the input for sequence logo when studying a sequence set, which is generally too simplified to represent all. One solution is to perform multiple sequence alignments (MSA) in advance, and use the gapped and aligned sequences as input to construct a single sequence logo, which has been supported by several tools. However, the problem with using MSA results is that it is difficult to indicate whether there are different patterns or motifs among sequences of different lengths. Let us take the B cell receptor (BCR) sequences as an example. Complementarity determining region 3 (CDR3) is the most hypervariable region in BCR and it is known that CDR3s of different lengths may have different affinities for certain antigens. Therefore, to discriminate the length of CDR3s when checking the motifs of CDR3s is essential for immune repertoire analysis. In addition to separately studying motifs for sequences of different lengths, we may also need multiple sequence logos for sequences of the same length but from different groups, which could be generated based on sample sources or clustering results. All of the above requires a convenient tool that allows researchers to take multiple sets of sequences as input, draw sequence logos synchronously and align them at the logo level to display pattern dynamics across different groups, so as to understand the sequence characteristics of the sample in a more delicate manner.

Besides CDR3s analysis, other motif-related studies, including transcript factor motif analysis, CRISPR array analysis, evolutionarily conserved sequences analysis and others, all have the same requirements.

To solve the problems, we developed MetaLogo, which satisfies the need to allow variable length or multi-group sequence as input and to perform multiple logo alignments, and provides researchers with figures in an aesthetic, multi-form, and highly customizable way.

Description

MetaLogo provides a public web server (locally deployable), and a stand-alone Python package at the same time to provide researchers with the most convenient service. Users can input files in *Fasta* or *Fastq* format, and specify grouping by length or by group id indicated in sequence names. MetaLogo draws a separate sequence logo for each group, and then performs alignment for multiple sequence logos in a local or global mode, according to users' choice.

Similarity metric

For each set of sequences, MetaLogo first calculates the information contents of amino acids or nucleotides at each position in bits [7]. In order to align different sequence logos, we need to measure the similarities between bit arrays of positions from different logos. For example, P and Q are bit arrays of positions from two different protein logos and defined as follows:

$$P = [p_1, p_2, \ldots, p_i, \ldots, p_n] \,,$$
$$Q \ = [q_1, q_2, \ldots, q_i, \ldots, q_n] \,,$$

where n is the number of amino acids types and item p_i andq_i represent the information contents of the i^{th} amino acid in the two positions, specifically. The arrays are sorted based on a fixed amino acids order.

To measure the similarity between P and Q, MetaLogo provides Dot Production (DP) and Cosine Similarity (COS) for users to choose from, which are commonly used as similarity measures and defined as follows:

$$DP(P,Q) \ = \textstyle\sum_i^n p_i * q_i,$$
$$COS(P,Q) \ = \frac{\mathrm{DP}(P,Q)}{Length(P)*Length(Q)},$$

where $Length(P)$and $Length(Q)$ represent the length of vector P and Q.

Besides bit arrays, we could also use frequency arrays to measure the similarity between positions. For each amino acid in one position, its frequency could be treated as the probability of one sequence having it in that position. Thus, here we could use similarity measurements designed for probability distributions.

MetaLogo allows users to choose the Jensen–Shannon divergence (JSD) [8] as the similarity measurement. The JSD is a method of measuring the similarity between two probability distributions, and is a symmetrized version of the Kullback–Leibler (KL) divergence [9]. Note in the following context, P and Q represent discrete probability distributions which sum to one. JSD is defined as follows:

$$JSD(P||Q) \ = \frac{1}{2} D_{KL}(P||M) \ + \frac{1}{2} D_{KL}(Q||M) \ ,$$

where $D_{KL}(P||M) \ = \ \sum_i^n P_i log \frac{P_i}{M_i}$, $D_{KL}(Q||M) \ = \ \sum_i^n Q_i log \frac{Q_i}{M_i}$, and $M \ = \ \frac{1}{2}(P + \mathrm{Q})$.

Bhattacharyya Coefficient (BC) [10] could also be used as a similarity measurement for two statistical samples. Since probability array does not indicate conservation like bit array do, hence MetaLogo provides an entropy (H) [11] adjusted Bhattacharyya Coefficient (EBC) as a choice to measure the probability array similarity, which is defined as follows:

$$EBC(P||Q) \ = \ BC(P||Q)\sqrt{(1 - \frac{H(P)}{H_{max}}) * (1 - \frac{H(Q)}{H_{max}})},$$

where $BC(P||Q) \ = \ \sum_{i=1}^n \sqrt{p_i q_i}$, $H(P) \ = \ -\sum_{i=1}^n P_i * log P_i$; $H(P)$ is the entropy of P and H_{max} is the max entropy for a n-dimensional probability vector.

Among these measurements, COS and KL consider both non-conservative and conservative patterns while DP and EBC only value conservative patterns among groups.

Alignment

The alignment between sequence logos is based on the Needleman–Wunsch algorithm, which is a classic global sequence alignment algorithm. When using MetaLogo, users can choose two alignment modes, one is pairwise alignments between adjacent sequence groups (Figure 1A), and the other is a global logo alignment among all sequence groups (Figure 1B). For global multi-logo alignment, MetaLogo adopts the method of progressive alignment construction [12]. The closest pair of sequence logos are aligned first, and the next logo closest to the aligned sequence logo set is successively added for alignment. Introduced gaps and inserts of each alignment are retained for subsequent alignments until all logos get aligned.

In the alignment process, users need to specify a certain similarity metric we mentioned above, and also the penalty for inserts and gaps. After padding and alignment, MetaLogo can visually highlight the highly similar pairs of positions between groups by connecting them using colorful strips.

Layouts

MetaLogo supports four different logo layouts, including horizontal, circular, radial, and 3D layouts. As shown in Figure 1 A-E, these diverse layouts are suitable for different scenes specifically. The horizontal layout is the default one, which can deal with most scenarios; the circular layout can more clearly show the conservations across multiple sequence groups; the radial layout is suitable to display sequences with conservative motifs in the middle or at the end of the sequences, rather than at the beginning; the 3D layout makes sequence logos more diverse and aesthetic.

MetaLogo allows customization of most of the operable elements in the figure, including figure size, ticks size, label size, labels, title, grids, margins between items, colors of items and so on. Users can also choose whether to display axis, ticks, labels, group ids, etc. Multiple formats of figures are supported, including *PNG*, *PDF*, *SVG*, *PS* and *EPS*.

Package Install and server deployment

Users can directly access our public web server (http://metalogo.omicsnet.org, Figure 1F), or install MetaLogo in Python package locally. Two examples of sequence sets are provided with the codes. One set contains sequences of *E. coli* transcription factor binding sites [13] (Figure 1 A-E; MetaLogo web server, example 1), the other set contains sequences of CDR3s of verified antibodies detected in BCR repertoires of individuals with COVID-19 [14] (See MetaLogo web server, example 2). A detailed tutorial for MetaLogo is provided online (https://github.com/labomics/MetaLogo/wiki). After the installation, users can run MetaLogo directly in the system terminal or import MetaLogo functions into their own scripts or projects. Users can also deploy MetaLogo as a web service on their local area network through Docker, which is convenient for people with no programming experience. Relevant parameters could be set for MetaLogo web server, including the number limitation of allowed sequences and the size limitation of uploaded files, etc.

Conclusion

MetaLogo is a new generator for aesthetic, customized and informative sequence logos. Unlike existing tools, MetaLogo can draw multiple sequence logos for sequences of different lengths or from different groups in one figure and perform alignment of sequence logos to reveal the pattern dynamics across groups. MetaLogo provides a free web server for public use, as well as a stand-alone Python package and a docker web service for local deployment. We will value the suggestions and comments from users, and continue to maintain code updates and upgrades to continuously contribute to the community.

Key points

- MetaLogo is a new sequence logo generator for variable-length sequences or multi-group sequences;
- MetaLogo performs pairwise and global sequence logos alignment to highlight the sequence pattern dynamics across different sequence groups.
- MetaLogo provides public web server, deployable local web server with docker, as well as stand-alone Python package for making highly customized sequence logos.

Funding

This work was supported by National Science and Technology Major Project grant [2018ZX10201001] and by the National Natural Science Foundation of China grant [31970567].

Conflicts of interest

The authors have declared no competing interests.

Acknowledgments

We thank colleagues in our lab including Pu Liu, Chao Feng, Sijing An and Runyan Liu for their careful reviews and feedbacks on the MetaLogo web server.

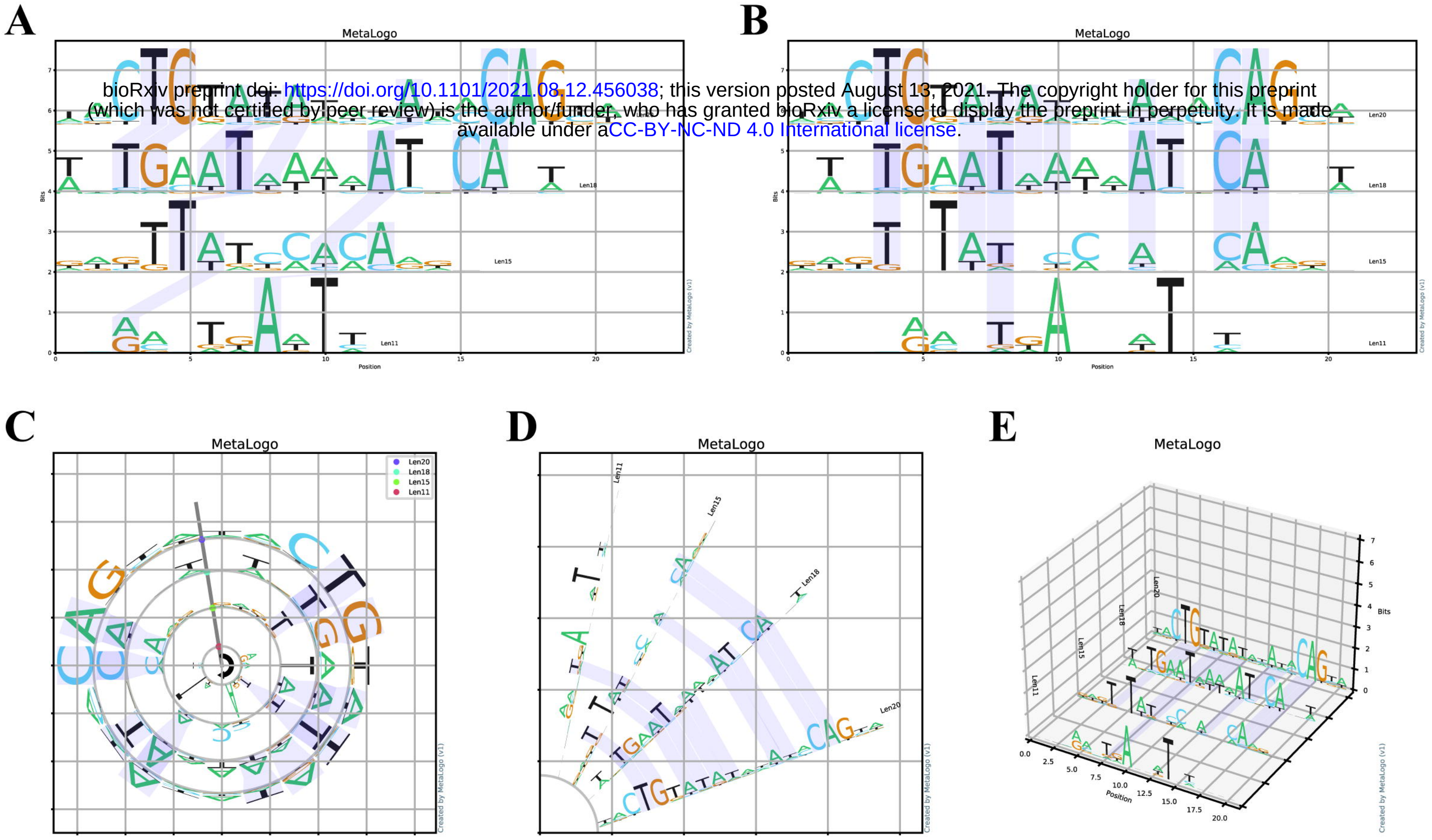
References

1. Schneider TD, Stephens RM. Sequence logos: a new way to display consensus sequences. Nucleic Acids Res 1990; 18:6097–6100
2. Crooks GE, Hon G, Chandonia J-M, et al. WebLogo: a sequence logo generator. Genome Res 2004; 14:1188–1190
3. Thomsen MCF, Nielsen M. Seq2Logo: a method for construction and visualization of amino acid binding motifs and sequence profiles including sequence weighting, pseudo counts and two-sided representation of amino acid enrichment and depletion. Nucleic Acids Res 2012; 40:W281–W287
4. Wagih O. ggseqlogo: a versatile R package for drawing sequence logos. Bioinformatics 2017; 33:3645–3647
5. Tareen A, Kinney JB. Logomaker: beautiful sequence logos in Python. Bioinformatics 2020; 36:2272–2274
6. Zheng L, Liu D, Yang W, et al. RaacLogo: a new sequence logo generator by using reduced amino acid clusters. Briefings in Bioinformatics 2021; 22:
7. Schneider TD, Stormo GD, Gold L, et al. Information content of binding sites on nucleotide sequences. J Mol Biol 1986; 188:415–431
8. Endres DM, Schindelin JE. A new metric for probability distributions. 2003;
9. Kullback S, Leibler RA. On Information and Sufficiency. The Annals of Mathematical Statistics 1951; 22:79–86
10. Bhattacharyya A. On a Measure of Divergence between Two Multinomial Populations. Sankhyā: The Indian Journal of Statistics (1933-1960) 1946; 7:401–406
11. Shannon CE. A mathematical theory of communication. The Bell System Technical Journal 1948; 27:379–423
12. Feng D-F, Doolittle RF. Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. J Mol Evol 1987; 25:351–360
13. Robison K, McGuire AM, Church GM. A comprehensive library of DNA-binding site matrices for 55 proteins applied to the complete Escherichia coli K-12 genome. J Mol Biol 1998; 284:241–254

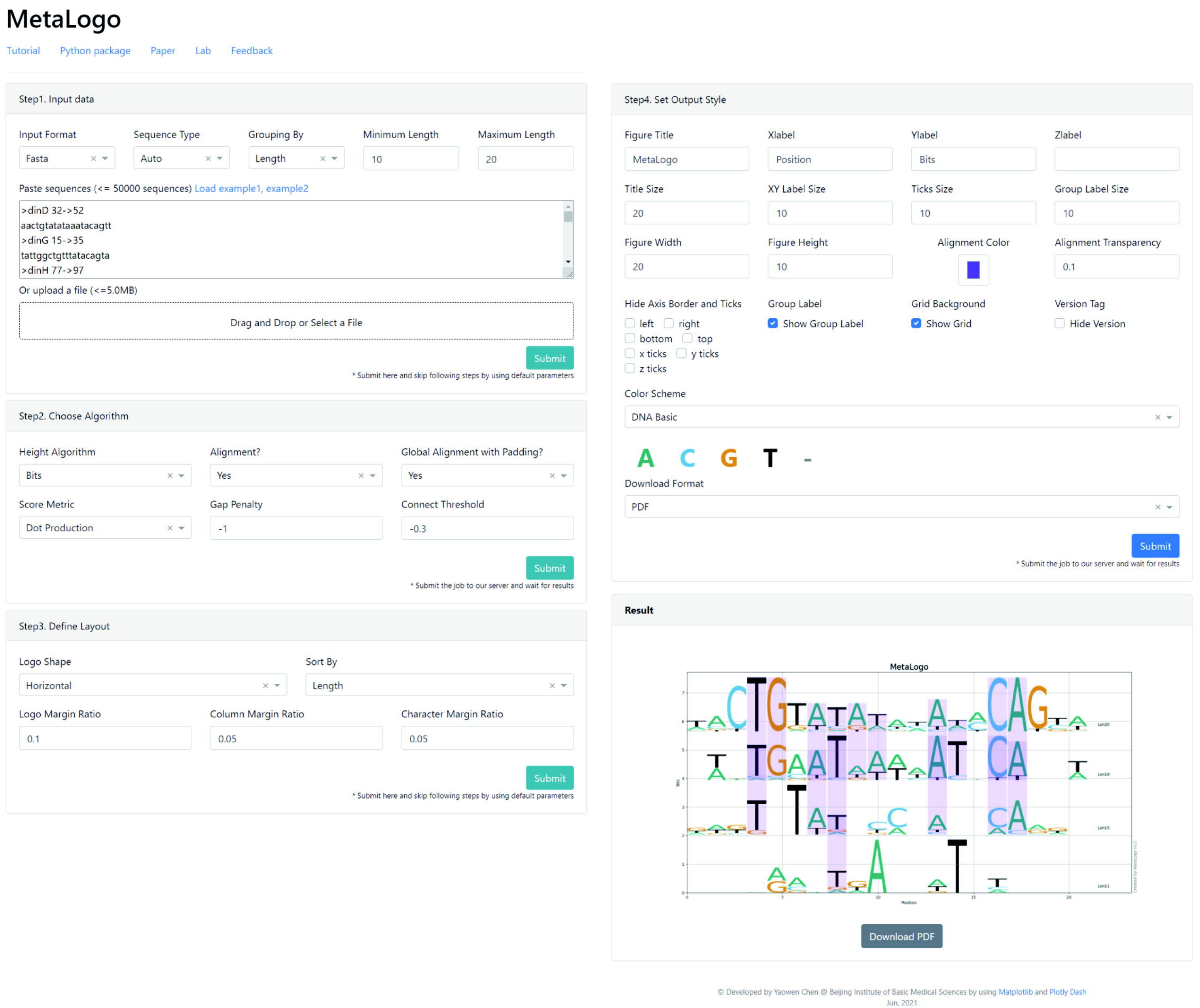
14. Montague Z, Lv H, Otwinowski J, et al. Dynamics of B cell repertoires and emergence of cross-reactive responses in patients with different severities of COVID-19. Cell Reports 2021; 35:109173

Figure Legends

Figure 1. Layouts, alignment modes and web server interface provided by MetaLogo.
A. Horizontal layout of MetaLogo under pairwise alignment mode. Conserved positions between sequence logos are indicated by grey bands. **B.** Horizontal layout of MetaLogo under global alignment mode. Sequence logos were filled with paddings according to a global logo alignment. Conserved positions among sequence logos are indicated by grey bands. **C.** Circular layout of MetaLogo under global alignment mode. **D.** Radial layout of MetaLogo under global alignment mode. **E.** 3D layout of MetaLogo under global alignment mode. **F.** Web server interface provided by MetaLogo.



F



MetaLogo

Tutorial Python package Paper Lab Feedback

Step1. Input data

Input Format: Fasta
Sequence Type: Auto
Grouping By: Length
Minimum Length: 10
Maximum Length: 20

Paste sequences (<= 50000 sequences) Load example1, example2

```
>dinD 32->52
aactgtatataaatacagtt
>dinG 15->35
tattggctgtttatacagta
>dinH 77->97
```

Or upload a file (<=5.0MB)

Drag and Drop or Select a File

Submit

* Submit here and skip following steps by using default parameters

Step2. Choose Algorithm

Height Algorithm: Bits
Alignment?: Yes
Global Alignment with Padding?: Yes
Score Metric: Dot Production
Gap Penalty: -1
Connect Threshold: -0.3

Submit

* Submit the job to our server and wait for results

Step3. Define Layout

Logo Shape: Horizontal
Sort By: Length
Logo Margin Ratio: 0.1
Column Margin Ratio: 0.05
Character Margin Ratio: 0.05

Submit

* Submit here and skip following steps by using default parameters

Step4. Set Output Style

Figure Title: MetaLogo
Xlabel: Position
Ylabel: Bits
Zlabel:
Title Size: 20
XY Label Size: 10
Ticks Size: 10
Group Label Size: 10
Figure Width: 20
Figure Height: 10
Alignment Color: 🟦
Alignment Transparency: 0.1

Hide Axis Border and Ticks
□ left □ right
□ bottom □ top
□ x ticks □ y ticks
□ z ticks

Group Label
☑ Show Group Label

Grid Background
☑ Show Grid

Version Tag
□ Hide Version

Color Scheme: DNA Basic

A C G T -

Download Format: PDF

Submit

* Submit the job to our server and wait for results

Result

Download PDF

© Developed by Yaowen Chen @ Beijing Institute of Basic Medical Sciences by using Matplotlib and Plotly Dash
Jun, 2021